

HD-A134 809

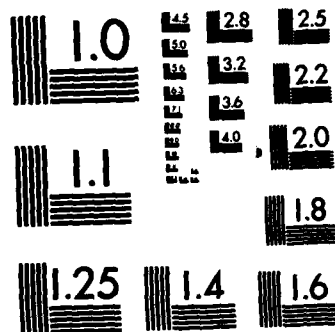
RESEARCH IN VLSI SYSTEMS HEURISTIC PROGRAMMING PROJECT 1/1
& VLSI THEORY PROJ. (U) STANFORD UNIV CA
J HENNESSY ET AL. NOV 83 MDA903-83-C-0335

UNCLASSIFIED

F/G 9/1

NL

END



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD-A134809

DEPARTMENTS OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
STANFORD UNIVERSITY · STANFORD, CA 94305



Research in VLSI Systems

Technical Progress Report

May 1983 - November 1983

Computer Science Department
Computer Systems Laboratory
Information Systems Laboratory
Integrated Circuits Laboratory

DTIC
SELECTED
NOV 17 1983
A

This document has been approved
for public release and sale; its
distribution is unlimited.

This work was supported by the Defense Advanced Research
Projects Agency, contracts MDA903-79-C-0680, MDA903-83-
C-0335, MDA903-80-C-0432, and MDA-903-80-C-0107.

The views and conclusions contained in this document are those of
the authors and should not be interpreted as representing the
official policies, either expressed or implied, of the Defense
Advanced Research Projects Agency or the U.S. Government.

DTIC FILE COPY

83 11 17 009

Research in VLSI Systems

Progress Report for May 1983 - November 1983

**Department of Electrical Engineering
Department of Computer Science
Stanford University
Stanford, California 94305**

**Heuristic Programming Project and VLSI Theory Project
DARPA Contract No. MDA903-80-C-0107
DARPA Order No. 3423**

**Principal Investigators: Edward Feigenbaum and Jeffrey Ullman
Computer Science Department
Monitored by R. Ohlander**

**Research in VLSI Systems
DARPA Contract No. MDA903-79-C-0680
DARPA Order No. 3773**

**Principal Investigators: John Hennessy and Thomas Kailath
Computer Systems Laboratory and Information Systems Laboratory
Monitored by P. Losleben**

**General Purpose VLSI-Based Multiprocessors
DARPA Contract No. MDA903-83-C-0335
DARPA Order No. 3773-6**

**Principal Investigator: John Hennessy
Computer Systems Laboratory
Monitored by P. Losleben**

**A Fast Turn Around Facility for Very Large Scale Integration (VLSI)
DARPA Contract No. MDA903-80-C-0432
Principal Investigator: James Meindl
Integrated Circuits Laboratory
Monitored by P. Losleben**

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER MDA 903-83-C 0335	2. GOVT ACCESSION NO. AD-A134869	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Research in VLSI Systems Heuristic Programming Project & VLSI Theory Project A Fast Turn Around Facility for Very Large Scale Integration (VLSI)		5. TYPE OF REPORT & PERIOD COVERED Technical Status Report May 1983 - November 1983
7. AUTHOR(s) J. Hennessy, R. Matthews, J. Newkirk, J. Shott, J. Ullman, H. Brown		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS Computer Science Dept.; Computer Systems, Information Systems, Integrated Circuits Laboratories; Stanford University, Stanford, CA.		8. CONTRACT OR GRANT NUMBER(s) MDA 903-83-C-0335 MDA903-80-C-0107 MDA903-80-C-0432
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Advanced Research Projects Agency Arlington, Virginia 22209		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Office of Naval Research (Stanford Branch) Stanford University Stanford, California 94305		12. REPORT DATE November, 1983
		13. NUMBER OF PAGES 40
		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Unclassified; approved for general distribution.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Very Large Scale Integration Computer Supported FTL Regular Expression Compilation Palladio: IC Designer's Assistant MIPS: A VLSI Processor Electron Beam Lithography Relative Layout Tools Micron CMOS Graphics Architectures Wafer Fabrication Facility		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Abstract

This report summarizes progress in the DARPA funded VLSI Systems Research Projects from May 1983 to November 1983, inclusive. The major areas under investigation have included: analysis and synthesis design aids, applications of VLSI, special purpose chip design, VLSI computer architectures, signal processing algorithms and architectures, reliability studies, hardware specification and verification, VLSI theory, and VLSI fabrication. The major research problems are introduced and progress is discussed; the Appendix contains a list of published research papers from these projects.

Key Words and Phrases: VLSI, design automation, computer-aided design, special purpose chips, VLSI computer architecture, signal processing, routing, layout, memory reliability, VLSI theory, knowledge-based design systems, IC fabrication.

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
Distribution/	
Availability Codes	
Avail and/or	
Dist	Special
A-1	



This work was supported by the Defense Advanced Research Projects Agency, contracts MDA903-79-C-0680, MDA903-83-C-0335, MDA903-80-C-0432, and MDA-903-80-C-0107.

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

Table of Contents

1 Design Description, Analysis, and Synthesis	5
1.1 Regular Expression Compiler	5
1.2 Pascal-to-Silicon Compiler	5
1.3 An Improved PLA Folder	6
1.4 TV - An nMOS Timing Analyzer	7
1.5 Control Compilation	9
1.6 Palladio: An Exploratory Environment for Circuit Design	10
1.7 Logic-to-sticks Conversion	11
1.8 Sticks Compaction	12
1.9 Control Description	13
2 VLSI Processor Architecture	14
2.1 MIPS - A High-Speed Single-Chip VLSI Processor	14
2.1.1 Recent progress	14
2.1.2 The Optimizing Compiler and Benchmarks	16
3 Theoretical Investigations	17
3.1 Funnel Pipelining and VLSI-Oriented Algorithms	17
3.2 A Time-Communication Tradeoff	18
3.3 VLSI Complexity of Coding	20
4 Fast Turn-Around Laboratory	20
4.1 Microlithography	20
4.1.1 MEBES Electron Lithography and Mask Making	20
4.1.2 Ultratech stepper	22
4.2 Processes, Devices, and Circuits	23
4.2.1 Fabrication of MIPS in 3.0 Micron nMOS	23
4.2.2 2.0 Micron CMOS Analog/Digital Gate Array	23
4.2.3 Plasma Etching of SiO ₂ Contacts	24
4.2.4 Deep Trench Etching	25
4.3 Interconnections and Contacts	26
4.3.1 Selective CVD of Tungsten	26
4.3.2 Exploratory Five Layer Aluminum Alloy Interconnection	27
4.4 Cell Library	29
4.5 Design Classes	29

Executive Summary

The major progress of note for this period is as follows:

1. *Regular Expression Compiler* A new coding scheme for nondeterministic states, called MCCC, has been put in place. It improves the area of PLA's generated in many cases, to the extent that the compiler now compares with hand designs in a mix of applications.
2. *MIPS: A VLSI Processor*. MIPS (Microprocessor without Interlock between Pipe Stages) is a project to develop a high speed (> 1 MIP) single chip 32-bit microprocessor. During this period, we received our chips from both MOSIS and the Stanford fabrication. Stanford provided testable 3μ parts first; MOSIS followed with fabrication-flawed parts at 3μ and a good fabrication run at 4μ . Testing on the Stanford parts uncovered one timing and one logical error; these results were verified on the 4μ MOSIS parts. These problems were corrected and we expect new chips back momentarily. Our new optimizing compiler was completed and it confirmed our design goal: MIPS is able to take better advantage of advanced optimizing compiler technology.
3. *TV: An nMOS Timing Analyzer*. TV and IA are timing analysis programs for nMOS VLSI designs. Based on the circuit obtained from existing circuit extractors, TV determines the minimum clock duty and cycle times. The recent additions to TV include work on the Interactive Advisor (IA) to support automatic timing optimization, and additions to TV to allow verification of hold times.
4. *PLA Partitioning* A practical and effective parallel partitioning algorithm for PLAs was developed. Several experiments using the algorithm were run. They showed average improvements of over 20% in area; the largest PLA's showing improvements up to 60%. The algorithm is relatively fast and accurate in estimating PLA overheads.
5. *Palladio: An Exploratory Environment for Circuit Design*. Palladio is an environment for experimenting with design representations, design methodologies, and knowledge-based design aids. During the past six months a prototype expert system which determines the gate sizes of transistors in an nMOS circuit was implemented and Palladio's logical reasoning-based simulator was refined and used to investigate a proposed supercomputer architecture.
6. *Computer Support - FABLE*. We have completed a prototype of a wafer fabrication description language called FABLE [Ossher 83] which will allow us to produce electronic run sheets which will guide a technician through the

fabrication sequence or, ultimately, control an automatic fabrication facility. A key feature of this language is the separation to the high level process step specification from the equipment-specific detailed execution of these high level steps to enhance the probability of a process specification in space or time.

7. *Parametric Testing Hardware.* We have installed a 10Mb/s ethernet link between our parametric test system and other DARPA VLSI computers on campus. Debugging of the link is in progress and down-loading of C test routines from a VAX 11/780 is being investigated.
8. *Electron Beam Lithography.* The Stanford MEBES machine has been used to routinely prepare masks for Fast Turn-Around Laboratory wafer fabrication including masks for two versions of MIPS with 3.0 μm minimum features.
9. *nMOS Wafer Fabrication.* The Fast Turn-Around Laboratory has completed fabrication of two versions of MIPS with 3.0 μm feature sizes.
10. *2 Micron CMOS.* We have developed a 2 μm mixed analog/digital CMOS gate array which includes poly- n^+ capacitors for switched capacitor filter applications. [Kuo-ISSCC 84]
11. *Deep Trench Isolation Technology.* We have continued an investigation of deep trench isolation techniques as a means of increasing the packing density of CMOS while reducing latch-up sensitivity. Initial electrical measurements of refilled structures indicate low values of fixed charge density Q_f for these structures.
12. *LPCVD Deposition of Tungsten.* Selective deposition of tungsten has been used as a contact metallurgy in both nMOS and CMOS processes.
13. *Sticks Compaction.* Supercompaction is a set of techniques to improve the predictability of 1-D sticks compactors. These techniques analyze a partially compacted cell and selectively move components or introduce jogs to break the critical path, thereby driving the compaction toward minimal pitch for the cell. Results so far indicate that Supercompaction applied to naively-drawn stick diagrams reduces the cell by 5-20% over straightforward 1-D compaction.
14. *Cell Library.* The 4-micron nMOS cell library is now available as a full-color book from Addison-Wesley.
15. *MEDIUM Tester.* The MEDIUM tester chip set is nearly complete, and 2 prototype testers have been built and debugged. Distribution of MEDIUM tester kits, including a PC board and the 5 MOSIS chips that comprise all of the tester electronics, will begin in early 1984.

16. *Two dimensional array layout* Finding an optimal upper bound on area and delay of configuration of two dimensional arrays into latices. This completes our earlier work on configuration of VLSI arrays in the presence of defects.
17. *Implementation of error correcting codes* Finding lower and upper bounds for area A and time T needed for VLSI implementation of error correcting coding circuits.
18. *Lower bound for matrix multiplication* Establishing lower bound for AT^2 for matrix multiplication as well as the class of l - independent functions. As a special case we have shown that a barrel shifter which can shift any sequence of length n to t bits must have $AT^2 \geq c.nt$.

Technical Progress

1 Design Description, Analysis, and Synthesis

1.1 Regular Expression Compiler

Anna Karlin designed and implemented a new technique, called "maximal clique compatibility classes," for selecting codes for the states of nondeterministic automata that we extract directly from the regular expressions. We work from the NFA, rather than converting to a deterministic version and coding its states in a conventional way, because the NFA structure has been found to provide some good clues as to the structure of the states, information that might be lost, or extractable only with great computational effort from the DFA.

A forthcoming paper summarizes the project, gives the details and motivation behind the MCCC state coder, and discusses examples that indicate the power of the MCCC method. An example of the power concerns a regular expression for pattern matching with 72 operands, that has a DFA with about 8,000,000 states (and therefore requires at least 23 bits). Previous methods had yielded codes for the NFA with 26--28 bits, while the MCCC method achieved a code with 24 bits. Moreover, because the NFA structure is retained, the number of terms in a PLA implementing the NFA is small, and the whole circuit is considerably smaller than a hand-designed PLA for the same problem.

Staff: A. R. Karlin, H. W. Trickey, J. D. Ullman.

References: [Karlin 83]

1.2 Pascal-to-Silicon Compiler

Howard Trickey is in the process of implementing a translator of a Pascal subset into silicon. The goal is to produce significantly better time/space tradeoffs than existing compilers. To do so, the data path is regarded as built from "resources" that can be either registers, limited arithmetic units, or busses. Initially, every program step has its own resources: a unit to perform the associated computational step, a register, if

necessary, to hold its result for future use, and busses to transmit its result where needed.

These resources are overlaid on one another, depending on the cost of combining them. For example, overlaying two busses results in a longer bus, which will likely still be cheaper than the two busses separately. Overlaying two addition steps has no associated cost, and overlaying an addition and subtraction is still likely to show a profit. However, overlaying an addition and a logical operation will probably not be a win.

Another optimization area concerns classical code optimization steps, where we are able to unroll loops and modify loops in other ways. Often, the system will find opportunities for parallelism in the unrolled code, but there is a time/space tradeoff that must be considered when we decide exactly how loops are to be treated.

Staff: H. Trickey.

Related Efforts: MacPitts (Lincoln Labs), Tseng and Siewiorek (CMU).

1.3 An Improved PLA Folder

Alan Siegel implemented a PLA folding routine that incorporates some novel features. First, it allows wires that are not paired, and wires in pairs need not match other pairs of wires. However, paired wires consisting of a signal and its complement are constrained to be adjacent, either on top or bottom, as is normal. The problem of testing legality of a folding is expressed as a "no cycles" condition in a graph.

The second novel feature is that, using standard PLA cells, should two wires in the same column each have taps on their last rows, a design-rule error results. Simple modifications of such cells still cause problems if both wires end in two taps. Obvious ways to avoid this rare-but-fatal condition are computationally expensive. The Siegel PLA folder avoids this special case with a "patch" that costs little in running time or space.

Staff: A Siegel.

Related Efforts: Hachtel et al. (IBM).

1.4 TV - An nMOS Timing Analyzer

TV and IA are timing analysis programs for nMOS VLSI designs. Based on the circuit obtained from existing circuit extractors, TV determines the minimum clock duty and cycle times. It calculates the direction of signal flow through all transistors before the timing analysis is performed, in contrast to combinations of designer-assisted and dynamic determination of signal flow, as in Crystal, being done at Berkeley. The timing analysis is breadth-first (block-oriented) and pattern independent, using only the values *stable*, *rise*, *fall*, as well as information about clock qualification. Its running time is linear in the number of nodes and transistors, and can analyze 4,000 transistors per minute of VAX 11/780 CPU time.

IA (TV's Interactive Advisor) allows the user to quickly experiment with ways to increase circuit performance. With the IA, the user can resize pull-ups and pull-downs or insert super buffers, and find out the effects of these changes on chip-wide performance interactively. By using information already computed by TV, it is able to propagate the effects of changes through 1,000 transistors per second of VAX 11/780 CPU time.

TV was heavily used in the MIPS project. When TV was run on the first version of MIPS, it predicted a cycle time four times longer than our original design goal. By making extensive modifications to the design we were able to reduce the cycle time to half the original prediction.

Accuracies within 20% for most critical paths compared to circuit simulation and fabricated chips have been achieved.

Since April several additions have been made to TV. First, the facilities of the Interactive Advisor (IA) have been greatly expanded. During the early summer an autopilot feature was added. In this mode the IA will suggest changes to the circuit and evaluate them by trying them out. It can propose and evaluate twenty changes in a 25,000 transistor chip in about one VAX 11/780 CPU minute.

Because D-type latches are used in nMOS design (instead of master-slave flip flops), signals can legally arrive (and correspondingly leave) a latch anytime the gating clock is high. The initial TV algorithm started and terminated all paths at rising edges of clocks, in order to simplify the analysis. Recently techniques have been developed and implemented which allow delays incurred while a clock is high to be charged to either that clock or the previous clock, so as to minimize and more accurately model the cycle time predicted for a design. These techniques still maintain the linear time of the previous algorithms.

Third, TV has been expanded to verify hold times. Two types of hold time checks are made. First, hold times are derived for all the inputs to a chip. Second, hold times are computed for all latches in the chip. If a user-specifyable safety margin is not met for latches within the chip, these latches are flagged as violations. This verification allows timing dependent two-phase clocking methodologies to be used (in contrast to strict two-phase designs which are guaranteed to work if the clocks are made slow enough), allowing for higher performance designs. These hold time checks also have running time linear in the number of nodes and transistors.

Finally, during the summer TV and the IA were readied for distribution to other universities and corporations. Extensive modifications were made to the code to improve the user interface, decrease the run time, use less memory, and handle a wider range of design styles. Included in this was the capability to analyze combinational designs and circuits clocked with asynchronous strobes. TV is currently being distributed by Stanford's Office of Technology Licensing for a nominal fee; contact Elizabeth Batson, Office of Technology Lincensing.

Staff: N. Jouppi

Related Efforts: Crystal (Berkeley)

References: [Jouppi 83a, Jouppi 83b]

1.5 Control Compilation

A major focus of our design aid work in this new contract will be the creation of a control synthesis system. We found this portion of the MIPS design to be a major stumbling block, both in complexity and in the difficulty of meeting the desired performance without careful hand decomposition and tuning. Our goal is to automatically synthesize optimized control implementations from high level specifications that go beyond the capabilities of our earlier system, SLIM [Hennessy 81].

Our initial attack has been on two key problems:

1. Decomposing PLAs into separate parallel PLAs. The first successes in this project are reported in [Hennessy 83]. The algorithm used is a merge style algorithm; it is reasonably efficient and accurate at estimating PLA costs. Improvements in the range of 20-40 percent of the original area are standard. Current work is focusing on placement estimation and time-based decomposition. We are also investigating an approach to the partitioning that involves incremental merging; this may yield better results than the algorithm proposed in [Hennessy 83] or the Berkeley Smile algorithm.
2. Developing alternative backends. In particular, creating a system to generate structured, multi-level logic implementations. We are currently exploring *optimized* Weinberger array implementations. Current systems for generating Weinberger arrays cannot compete with PLA implementations. Our goal is to generate a Weinberger backend that will be more efficient than PLAs in some important cases. Some initial progress in exploring the use of simulated annealing to solve placement problems has been made.

Staff: C. Rowen, J. Hennessy, Y. Brandman, A. El Gamal

Related Efforts: Smile (Berkeley and IBM, Yorktown Heights), Lincoln Boolean Synthesizer (Lincoln Labs),

References: [Hennessy 83]

1.6 Palladio: An Exploratory Environment for Circuit Design

Palladio is an environment for experimenting with design representations, design methodologies, and knowledge-based design aids. It differs from other prototype design environments by providing the means for constructing, testing and incrementally modifying or augmenting design tools and design languages.

Palladio provides a testbed for investigating elements of circuit design that includes specification, refinement, simulation, and use of existing designs. It has facilities for conveniently defining models of circuit structure or behavior. These models, called perspectives, are similar to design levels; the designer can use them to interactively create and refine circuit design specifications. Perspectives can include rules that constrain how circuit components may be composed in that perspective to form more complex components. Palladio provides an interactive graphics interface for displaying and editing structural perspectives of circuits in a uniform manner and a declarative logic behavioral language with an associated interactive behavioral editor for specifying a design from a behavioral perspective. Further, a generic, event-driven simulator can simulate and verify the behavior of a circuit specified from any behavioral perspective and can perform hierarchical and mixed-perspective simulation. Facilities are available for conveniently creating and using prototype libraries. The entries in a prototype library are components of arbitrary complexity, specifiable from multiple perspectives.

During the past six months we have completed a prototype knowledge-based, expert system design refinement aid which determines the gate sizes of the transistors in an NMOS circuit. The system is interfaced with a previously implemented expert system for assigning mask levels to interconnect, and it takes into account global speed and power goals, constraints and trade-offs. Also, we have used Palladio to investigate various message passing schemes for a proposed multi-processor, message-based super computer architecture.

We are continuing our work on the design of a language that spans the spectrum of functionality, behavior and structure thus eliminating some of the parallel specification languages currently required; and of a language in which circuit design problems (and

theories of circuit design) can be stated, based on the assumption that the circuit design problem and the circuit design co-evolve. Basic terms in such a language include design goals, tasks, constraints and tradeoffs.

We have recently started a collaborative effort with the Fairchild Laboratory for Artificial Intelligence Research to implement the basic framework underlying Palladio on the Symbolics 3600 computer. This framework will serve as a common implementation environment for several circuit-related research activities at Stanford and Fairchild. These activities include design specification, design verification, simulation, diagnosis, and test generation. In particular, the resulting system will be used to investigate various architectures for supercomputers.

The Palladio system is described in detail in [Brown 83].

Staff: H. Brown, G. Foyster, N. Singh (Stanford and Fairchild), C. Tong, J. Yan.

References: [Brown 83, Yan 83]

1.7 Logic-to-sticks Conversion

Dumbo is a program aimed at directly laying out random logic from logic diagrams. It targets its output to stick diagrams for compaction by our sticks compactor, Lava. The motivation for a tool of this sort is to ease the layout of miscellaneous logic, especially control logic, in a design. Much logic of this sort is not area-critical, but its design and layout can consume a lot of time using standard techniques.

We have now refined Dumbo to the point where it produces layouts feasible for some miscellaneous logic. For small cells (under 25 components), Dumbo's initial layout will be at most about 2 times larger than one derived from hand-drawn sticks; with hints, this penalty can easily be reduced further. For larger cells, the penalty can become much larger, but is yet easier to reduce using hints.

However, Dumbo still experiences considerable area inefficiency due to the sensitivity of sticks compactors to the vagaries of a particular stick diagrams that it produces. Rather

than trying to improve the quality of stick diagrams that Dumbo produces, we have turned our attention to improving the quality of the sticks compactor itself (see below).

Staff: W. Wolf, R. Mathews

References: [wolfCIT83 83]

1.8 Sticks Compaction

One-dimensional sticks compactors are sensitive to the details of the stick diagrams that they are given to compact. Two topologically equivalent stick diagrams can produce very different compacted cells. The essential reason for this behavior is that the compaction algorithms cannot properly exploit the degrees of freedom present in the stick diagram to prevent components from locking against each other during compaction.

The problem that we are investigating is how to compact a stick diagram to achieve a pitch specification in a specified direction. In the majority of a layout, the designer is typically not trying to minimize cell area per se; rather, he is trying to minimize cell area subject to meeting a particular pitch specification in one dimension. Therefore, we are seeking techniques to guide the compactor toward a solution with the minimum pitch in a specified direction, increasing the predicability of the results of compaction by forcing the compactor toward the same solution irrespective of the details of the initial stick diagram. The resulting compaction scheme is called Supercompaction.

To date we have investigated two principal Supercompaction techniques: moving components apart to break constraints between them, and introducing jogs into the stick diagram. Both optimization techniques work by analyzing the critical path in a partially compacted cell and rearranging components or introducing jogs to break the critical path. Naturally, these manipulations cause the cell to grow in the direction perpendicular to the preferred direction as well as reducing the pitch in the preferred direction.

We have investigated a few variants of these techniques by comparing compaction

results for cells drawn from a variety of different sources and compacted using the standard Lava compactor and the Supercompactor. Our early results indicate that while supercompaction performs no better (and sometimes worse) than simple compaction for carefully optimized stick diagrams, it easily achieves 5-20% pitch reductions over simple compaction for naively drawn stick diagrams. Over a test suite of 10 cells, 5 were smaller when Supercompacted.

These initial results are in keeping with our goal of developing a predictable compactor. Initial results for jog introduction are even more promising, and we are continuing to investigate supercompaction techniques.

Staff: W. Wolf, R. Mathews, D. Perkins

Related Efforts: CABBAGE (UCB), other 1-D and 2-D compactors

References: [wolfICCAD83 83, lavaCOMPCON 82]

1.9 Control Description

Plunder is a new control-description language that we have investigated as an alternative front end to our control synthesis systems. The Plunder language is essentially the control portion of the C programming language. Thus, the designer does not need to describe his control sequences as FSM state diagrams; rather, he can write in familiar programming-language control structures. On the other hand, he sacrifices the fine control over the structure of the state machine that a language such as SLIM provides.

Plunder has been used by students in the Stanford design classes. Their experiences suggest that while most of the software control notions carry over to hardware, there are important differences that the ultimate language of this sort must cater to. In particular, the designer often must know precisely what actions are occurring on which clock cycle. Also, description of concurrent activities is an immediately pressing problem for a hardware control language. Nevertheless, Plunder enforces a structuring on control descriptions that generally eases that portion of the IC-design task.

The following is the cononical control-description example, the traffic-light controller:

```
#define green 0
#define yellow 1
#define red 2

input ts, tl, cars;
output restart, hl[2], fl[2];

fsm traffic ()
{
    restart;
    do hl=green; fl=red; while (~tl | ~cars);
    restart;
    do hl=yellow; fl=red; while (~ts);
    restart;
    do hl=red; fl=green; while (~tl & cars );
    restart;
    do hl=red; fl=yellow; while (~ts);
}
```

Staff: D. Perkins

Related Efforts: SLIM(SU), MacPitts(LL)

2 VLSI Processor Architecture

2.1 MIPS - A High-Speed Single-Chip VLSI Processor

MIPS (Microprocessor without Interlock between Pipe Stages) is a project to develop a high speed (> 1 MIP) single-chip 32-bit microprocessor. Like the RISC project at Berkeley, MIPS uses a simplified instruction set and is a load-store architecture.

The MIPS architecture is summarized in previous technical progress reports and is discussed in several publications.

2.1.1 Recent progress

The project history since March has been as follows:

March 19: The design was submitted to MOSIS for fabrication using their 3μ and 4μ feature size nMOS runs.

- April 28: Fabrication began at the Stanford Fast Turn Around Facility on their 3μ feature size process.
- June 21: The Stanford line finished 8 wafers of 81 die each. Unfortunately, an implant problem resulted in enhancement and depletion thresholds that were about a volt too high. This made testing difficult but not impossible. The design faults were eventually isolated with these chips.
- June 24: Ten 3μ feature size parts arrived from MOSIS. They had been done with an experimental process and none of them worked.
- July 6: After a two week delay to set up the testing hardware, power was first applied to the design. Initial success was slow in coming as the threshold problems of the Stanford run were dealt with. The application of -5V of substrate bias yielded the best results.
- July 16: The design had been shown to be mostly working but was acting oddly in some circumstances. A timing error explained the strange behaviour. The error caused the instruction register to be latched at twice the desired frequency.
- July 20: During the final stages of testing of the design, a minor logic problem was uncovered that prevented access to one bit of processor state.
- July 26: The 4μ feature size parts arrived from MOSIS. They corroborated the previous results of the Stanford run, and included the first known part with no fabrication defects.
- August 14: A second iteration of the design was submitted for fabrication to the Stanford facility.

The excessive latching of the instruction register is caused by qualifying a signal generated on ϕ_1 by ϕ_1 . This causes glitches at the beginning of the next cycle. We were well aware of these glitches while designing the chip and were careful to nullify their unwanted effects in all but this one case. The simulator did not catch the problem because of its inherent timelessness, and our informal audits of these glitches missed the bug because the signals were qualified in the MPC and used in the IDU: two pieces in different spheres of influence. Unfortunately we did not have available to us a timing simulator that was reliable enough to be worth using.

The other undetected bug made one particular bit of state unreadable. This is a matter of sloppiness and excessive trust in the work of the other members of the team. This particular bit was a thorn in our side for a long time due to its irregularity. Its implementation was carefully thought out, done and reviewed several times. When it came time to simulate it, however, the informal submission deadline was drawing near and more interesting things were still not simulated. A couple of quick tests appeared to show it functioning properly and it was summarily written off as working. The random test generator could not test any of the exception hardware because of the intervening instruction level simulator.

To circumvent the inability of the Medium Tester to do tests at speed, we have undertaken to build a special purpose board to determine the maximum performance of the processor. A multibus board with 64K bytes of two-ported fast static RAM, the MIPS processor and some clock generation circuitry will be inserted into a SUN workstation [BaskettBechtolsheim 82]. The M68000 will be able to load the memory with a program which the MIPS processor will then be able to execute. The clock generation circuitry will allow the M68000 to vary under program control the cycle time, the duty cycle and skew time of all the clocks.

Currently we are awaiting the return of the corrected design from the Stanford fabrication facility. In addition, we submitted a revised design (MIPS 2.1) that uses a new bus structure; we believe (based on TV measurements) that the new design will achieve our speed goals on a 3v run.

2.1.2 The Optimizing Compiler and Benchmarks

We have recently completed the integration of the MIPS code generator with our UCode global optimizer. The results have been extremely rewarding in several ways. First, the optimizer performs quite well and enhances the performance on a set of kernel benchmarks by an average of almost 60%. Second, this average performance improvement exceeds the average improvement obtained on several other machines (The S-1, the DEC-10, and the M68000) by an average of about 15%. This confirms our initial design goal of designing the architecture as a good compiler target. We have also shown

that a relatively small number of registers (11-12) are needed to support the active variables within a procedure.

Our continuing plans involve a series of instrumentation and measurements steps using the MIPS compilers. *Staff:* J. Gill, T. Gross, J. Hennessy, N. Jouppi, S. Przybylski, C. Rowen.

Related Efforts: RISC (UCB), IBM 801 (IBM Yorktown), Cray-II (Cray Research).

References: [HennessyJouppi 81, HennessyJouppi 83, HennessyGross 83, GrossThomas 83, Przybylski 83]

3 Theoretical Investigations

3.1 Funnel Pipelining and VLSI-Oriented Algorithms

A forthcoming paper discloses two techniques developed by Peter Hochschild, Ernst Mayr, and Alan Siegel for solving graph problems such as minimum spanning trees and biconnected components in (roughly) linear space and time in a VLSI environment. Their algorithms use a tree organization like those of Lipton and Valdes (1981 *IEEE Symp. on Foundations of Computer Science*) but unlike the latter, the new algorithms allow the input (an adjacency matrix) to be read only once, in an input schedule that is data independent.

The two new techniques used are called "filtration" and "funnel pipelining." Filtration is a technique used to discard irrelevant input data rapidly. A funneled pipeline is built from a series of increasingly thorough filter stages. Transition times along such a pipeline of filters form an exponentially increasing sequence of delays, but the increase in delay is exactly balanced by an increasing degree of filtration. That is, the i^{th} filter takes time 2^i , but each filter produces only half as much output as it takes input. Thus, the total time spent by each filter is the same, and the whole system represents an effective use of parallelism.

Staff: A. Siegel.

Related Efforts: Lipton-Valdes (Princeton).

References: [Hochschild 83]

3.2 A Time-Communication Tradeoff

Most results on the limitation of our ability to compute on a chip concern area-time tradeoffs, like the AT^2 results of Clark Thompson and others. Another limitation to our ability to compute in silicon, or by networks of microprocessors, concerns a trade between time and the amount of communication that must go on, either across a chip or between chips. In particular, we have shown a surprising result about systems that compute values with dependencies in a square grid. That is, each value requires for its computation the value below and the value to its left. Examples of problems normally solved this way include many numerical problems, like taking derivatives, and the calculation of longest common subsequences of two symbol strings.

If this grid is n by n , and we share the job among k processing elements, then the time required is at least proportional to n^2/k . Further, if we measure communication to be the number of values computed by one processor and used by another, then no matter how we divide responsibility for the values among the k processors equally, the communication c must be at least $nk^{1/2}$. Moreover, these bounds are individually the best possible, since there are algorithms that meet the bounds for any number of processors up to n .

However, we can prove that the two bounds cannot be met simultaneously, thus exposing a subtle limit on our ability to implement algorithms of this type in VLSI. Specifically, we show that no matter how the values are assigned to processing elements (even unequally), the product ct must be at least n^3 . Note that this result is stronger than applying the trivial bounds, which only say that ct must be at least $n^3/k^{1/2}$.

Staff: J. Ullman.

References: [Ullman 83]

In [GreeneElGamal 83], Greene and El Gamal investigated the problem of connecting interchangeable computing elements on a large, partially defective integrated circuit into a fully functional systolic array. (The technologies developed for redundant 64K RAM chips provide the means for programming the connections.) Lower bounds on the area required for the interconnect, and on the maximum wire length, were proved for several array configurations. For the cases involving linear arrays, these bounds were shown to be tight to within a constant; linear-time algorithms for specifying the appropriate connections were given. However, no connection scheme attaining the lower bounds for a two-dimensional array was obtained, though a result of Leighton and Leiserson came close.

More recently, Greene and El Gamal [Greene 83] have developed a linear-time algorithm for configuring a two-dimensional array that uses wiring area and wire length within a constant of the lower bounds. The algorithm is based on finding flows in networks with random capacities, the capacities being determined by the defect states of the circuit elements. Somewhat surprisingly, the fraction of circuit area devoted to wiring can be held constant as the size of the array grows. The maximum wire length must grow at a moderate rate, proportional to the square root of the logarithm of the number of array elements. The yield loss due to random defects approaches zero. As with the one dimensional problems, there is a tie-in with percolation theory. This promises to be of use in extending the results to accommodate defective wiring, as well as defective elements.

Staff: J. Greene, A. El Gamal

Related Efforts: Work of Leighton and Leiserson, MIT

References: [Greene 83, GreeneElGamal 83]

3.3 VLSI Complexity of Coding

In [ElGamalGreene 83] it is shown that the area A , computation time T , and pipeline period P for any integrated circuit that encodes or decodes an (n, Rn, t) binary t -error-correcting code must satisfy the lower bound $AT^e \geq AP^e = \Omega(R^2 nt)$ under a general VLSI model. This bound also holds for an (n, Rn, P_e) -code if t is replaced by $\log(n/P_e)$, where P_e is the average probability of decoding error. It is also shown that any circuit that computes an l -independent function (as defined by Grigoryev) with n inputs and Rn outputs must satisfy $AT^e \geq AP^e = \Omega(R^2 nl/(1+R))$.

An encoder for linear codes can be implemented using a circuit with $A = O(Rn^2 \log Rn)$ and $T = O(\log(Rn))$. For Low-Density Parity-Check Codes, $R = \Theta(1)$ and $t = \Theta(n)$ so $AT^e \geq AP^e = \Omega(n^2)$; the decoding algorithm of Zyablov and Pinsker for a subclass of these codes can be implemented on a circuit with $A = O(n^2 \log n)$ and $P = O(1)$, or $A = O(n^2)$ and $T = O(\log n)$. For binary primitive BCH codes of constant rate, $AT^e = \Omega(n^2 / \log n)$; a decoding algorithm can be implemented on a circuit with $A = O(n^2 \log^2 n)$ and $T = O(\log^2 n \log \log n)$.

Staff: J. Greene, A. El Gamal, and K. Pang

References: [ElGamalGreene 83]

4 Fast Turn-Around Laboratory

4.1 Microlithography

4.1.1 MEBES Electron Lithography and Mask Making

MEBES was used during this period to write masks for the Ultratech 1:1 stepper, the Canon 4:1 aligner, and other research jobs; and to write on wafers for multi-level resist and metallization tests.

Mask sets for the Ultratech 1:1 stepper included 3 versions of MIPS (3.0 μm nMOS), a CMOS multi-project set with 2 each of 3 primary die, and a reticle to generate calibration wafers for a laser scanning monitor to record surface defects down to 1 μm on wafers.

Software was written on a VAX 11/750 (Glacier) to layout reticles for the Ultratech and produce all the required MEBES job files for either a nMOS or CMOS run. Differences in mask formats between the Ultratech 900 stepper and a Perkin/Elmer full field projector make it difficult to utilize the job file preparation software in use at MOSIS. Software has been written in C to perform the following tasks in MEBES job deck preparation. All of the primary die within a "generic" Ultratech field are positioned. Die positions can be specified in absolute field coordinates or relative to another die. All the other patterns, such as field keys and targets (for auto-alignment on the Ultratech), optical alignment targets, registration verniers, linewidth control patterns, and either nMOS or CMOS test stripes, which are needed within a field, are dropped into the field as required. The output of this program is a text file which includes all the position coordinates within a field for all die for this particular run.

Two other files have been generated specifying parameters for either a nMOS or CMOS run to be fabricated at Stanford. Parameters included for each mask layer of a process are:

1. Mask titles and extensions
2. Tone and sizing (bloats and shrinks) for primary die
3. Position, tone, and linewidth of this level's field keys (for Ultratech auto-alignment)
4. Vernier positions (to evaluate Ultratech auto-alignment accuracy)

The die coordinate file and one of the nMOS or CMOS parameter files are then used to generate the actual MEBES job files, taking required pattern die names from a library of existing patterns such as all alignment die. The job files are then transferred to MEBES for writing of the 1x reticles and preparation for use in the Ultratech.

Masks which have been written on MEBES for the Canon FPA-141 include CMOS defect array, five-layer metal, and linewidth test structures, and analog-digital CMOS gate arrays. Other jobs included liftoff metallization tests and masks for "Brush fire lithography" where only the edges of features are written and their interiors are filled in by selective etching.

Wafers were written for registration and proximity effect tests, metallization tests, and resist evaluations which included 0.125 μm lines and spaces written with a single address in PMMA resist using the LaB_6 source.

Work is underway on an 10 Mb/s ethernet link between MEBES and a VAX 11/750 (Glacier). Interface cards have been installed in both machines and connected via an ethernet cable. Software is now being written for the Data General computer which runs the MEBES.

The contract has been signed with P.E. EBT for "1/8th micron" performance of the MEBES. Work under this contract will improve the placement accuracy of the MEBES to correspond with the 1/8th μm spot size of the LaB_6 gun.

4.1.2 Ultratech stepper

An interface has been successfully established between the Ultratech 900 stepper and the VAX 11/750 (Glacier) to facilitate data transfer. This interface allows the reticle data, which provide the necessary stepping information for each reticle, to be stored in Glacier disk files instead of magnetic tape cassettes. As a result, data management, archiving and retrieval are much more efficient than can be achieved by a stand-alone Ultratech stepper. Further advantages of such an interface would include the constant and automated monitoring of the stepper performance such as alignment accuracy. A C-language program was also developed for the generation and subsequent processing of this reticle data. Its implementation on Glacier not only eliminates the need for using the dedicated digital controller on the Ultratech (a HP 9825 calculator) for reticle data handling, but also expedites and simplifies this task.

An autoloader was installed on the Ultratech to reduce wafer handling by the operator and thereby improve the yield. A newer version of the machine operation software, in conjunction with several retrofitted hardware modifications, has significantly enhanced the machine throughput. Corresponding modifications on the Ultratech-to-Glacier interface and the reticle data management software have also been completed.

Using reticles generated on the MEBES and covered with pellicles, the Ultratech stepper

has served as the photolithographic tool for four nMOS and one CMOS VLSI runs. Resolution down to 1 μm has been demonstrated during these runs and, with programmable image offsets, the alignment errors have been consistently contained within 0.1 μm with the variation typically less than 0.1 μm at one sigma.

Staff: R. F. W. Pease, D. Dameron, C-C. Fu, E. Crabbe.

4.2 Processes, Devices, and Circuits

4.2.1 Fabrication of MIPS in 3.0 Micron nMOS

During the present report period we have completed two runs of MIPS using 3.0 μm nMOS technology which results in a die size of 5.4 mm by 5.6 mm. Although the first run received incorrect threshold shift implants which required operation with substrate bias, the first version of MIPS was largely functional as detailed in other sections of this document. The results of this fabrication run resulted in two new versions of MIPS: a corrected design and a new design featuring higher performance bus drivers. These two versions were assembled on a single set of Ultratech reticles. Because low defect density is crucial in a design of this complexity, we sent each of these reticles to Master Images for mask inspection using a KLA-100 mask inspection system. Fabrication of this set of wafers is complete and parametric testing is in progress.

4.2.2 2.0 Micron CMOS Analog/Digital Gate Array

Our 2 μm process has been modified to include provision for high-quality MOS capacitors as would be required for switched capacitor filter applications. An additional n^+ implant is used early in the process sequence to produce the lower electrode of a MOS capacitor with a low voltage coefficient of capacitance. Electrical characterization of a switched capacitor filter using this process is in progress. [Kuo-ISSCC 84] Because the n^+ and p^+ source/drain regions in this process are quite shallow (0.3 μm and 0.55 μm , respectively) we have incorporated a selective deposition of tungsten in the contact regions to prevent junction leakage problems with the sputtered aluminum allow interconnections.

4.2.3 Plasma Etching of SiO_2 Contacts

The reduction of contact windows below $4\text{ }\mu\text{m}$ has made plasma etching of contacts essential. In order to insure reliable contacts the plasma etching process is being studied in terms of polymer formation, wall slope, contact resistance, and end point detection.

Previously we reported the use of a high polymer forming etching process with selectivities of SiO_2 to Si of greater than 10 to 1. Unfortunately, this process was found to be uncontrollable when applied to device wafers. By backing off on the polymer forming agent (CHF_3) a controllable process was obtained with a selectivity of 5 to 1 which was found to quite adequate. The conditions for this process which is performed in a Branson/IPC Sigma 80 etcher are: 2.8 slm of He, 293 sccm of C_2F_6 , 175 sccm of CHF_3 , 10 torr pressure, and 800 watts of rf power. The etch rates are 100 Angstrom/sec for thermal oxides and 150 Angstrom/sec for 8% P-glass. The walls are near vertical for this process, and can be sloped by adding O_2 to erode the resist at same time the oxide is being etched. The addition of 80 sccm of O_2 to the above process results in a wall slope of 50 degrees which can easily be covered during subsequent metalization step.

Unlike wet contact etching, plasma etching can result in residual layers being left behind after the etching process. These layers can significantly increase the contact resistance of a device. We are currently investigating the sources of these layers and means of eliminating them. A strong suspect are reactions between dopants, such as As and B, and reactants in the plasma.

With the low selectivity of plasma oxide etching and the shallow junction depths of current devices, excellent end-point detection is needed for this process. Unfortunately, the usual end-point detection scheme, optical spectrum emission, suffers from low signal-to-noise ratios because of the small area being etched for a typical contact etch mask. As an alternative we have begun investigating the monitoring the DC current through the open contact holes. Initial result look very promising in that the DC current shows a significant change as the contact holes open up. Work is in progress to eliminate alternative DC current paths in the plasma. It should noted that current levels and voltages are such that no damage should occur to the p-n junctions in the current path.

4.2.4 Deep Trench Etching

Work has continued on developing deep trench isolation for the elimination of latch-up in CMOS. During this period this effort has concentrated on improving the trench profile in order to eliminate voids left from incomplete refill. In addition an investigation of parasitic channels was begun.

Trench etching using fluorine based chemistry (C_2ClF_5/SF_6) in the plasma mode (wafer on grounded electrode) results in "U" shaped trenches with 0.2 to 0.3 μm sidewall bow. When these trenches are filled in using highly conformal LPCVD poly Si, the side wall bow results in voids near the top of the trenches. A two step etching process consisting of an initial isotropic etch (3000 Angstrom deep) followed by the 5 μm anisotropic etch was found to eliminate this bow. The penalty for this process is a 0.25 μm undercut per side. Using this dual step process, refill without voids has been demonstrated.

A principal problem with reported trench isolated CMOS devices has been parasitic channels in the n-channel devices. These channels are believed to be caused by high values of fixed interface charges, Q_f , associated with the growth oxide on the walls of the trench. To investigate these channels, a mask set was designed to measure Q_f on the bottom and sides of our trenches. Using a 700 Angstrom gate oxide grown on the walls of the trench and a 3000 Angstrom thick doped poly Si gate deposited on the walls, initial results indicate a low 10^{10} per cm^2 fixed charge density. This Q_f value is significantly below reported values and is probably due to lower ion energy associated our use of "plasma" mode etching as opposed to the reported trench etching using the higher ion energy RIE mode etching.

Staff: J. D. Shott, J. P. McVittie, J. R. Pfister, K. C. Saraswat, S. H. Goodwin, L. Lewyn, J. D. Plummer.

Related Efforts: Oldham (Berkeley).

References: [Pfister-Maui 83, Moslehi-Maui 83, Goodwin-Maui 83, Pfister-ISSCC 84, Lewyn-ISSCC 84]

4.3 Interconnections and Contacts

With advances in integrated circuit technology, device dimensions are being scaled down and the chip size continues to increase. The smaller size of the intrinsic device makes it faster, however, parasitics extrinsic to the device, e.g., contact resistance and interconnection resistance can overshadow the performance. In order to minimize deleterious effects due to contact resistance and to fully exploit the potential packing density of VLSI, our research in the areas of interconnections and contacts has focused on (a) the selective CVD of tungsten as a contact metallurgy in MOS technology and the use of aluminum alloy/planarized SiO_2 multi-level interconnections.

4.3.1 Selective CVD of Tungsten

Selective low-pressure chemical vapor deposition of tungsten (W) has been investigated has been investigated as a contact metallurgy for shallow n^+ and p^+ junctions. Depositions have been studied in an ambient of $\text{WF}_6 + \text{H}_2$ in a hot-wall furnace at temperatures from 275 deg C to 450 deg C and at pressures from 0.2 to 1.0 torr. Under proper conditions W has been selectively deposited onto Si, PtSi, and Al surfaces through contact windows as small as $1.25 \mu\text{m}$ by $1.25 \mu\text{m}$. Encroachment of W underneath SiO_2 along the contact interface has been eliminated by optimizing the deposition parameters. Deposition on Si was found to occur by a combination of Si and H_2 reduction of WF_6 , resulting in a consumption of about 200 Angstrom silicon. Deposition on PtSi and Al occurred due to H_2 reduction of WF_6 .

W contacts have been made to phosphorus and boron doped diffusions by first selectively depositing W and then evaporating Al. Contact resistance has been measured as a function of doping density. At a doping density of 10^{20} cm^{-3} the contact resistance to n^+ diffusion was about $10^{-8} \Omega\text{-cm}^2$ and to p^+ diffusion about $5 \times 10^{-7} \Omega\text{-cm}^2$. W has been found to be a good barrier against Si diffusion in Al. Using this technology, contacts have been made to MOS transistors with junction depths as shallow as $0.3 \mu\text{m}$. Schottky barrier diodes of large area have been fabricated by selectively depositing W on n-type Si. Extremely reproducible characteristics were obtained because of the cleanliness of the W-Si interface. Ion beam induced formation of WSi_2 by first depositing W on Si and then ion implanting As or Si is currently being investigated.

This technique appears to have good potential for selective silicidation of source, drain, and gate regions of a transistor. Since W is selectively deposited on Si and not on SiO_2 , the problem of *bridging* between gate and source/drain regions be eliminated.

Selective deposition technology is also being investigated as a means of planarizing the vias in a multi-layer metal system. Vias with vertical sidewalls will be etched and then refilled by selectively depositing W in the contact regions resulting in a planar surface. If the thickness of the selective deposition can be increased sufficiently the step coverage during subsequent Al alloy depositions can be greatly improved.

4.3.2 Exploratory Five Layer Aluminum Alloy Interconnection

A study of aluminum, aluminum/copper (with and without silicon), and aluminum/titanium has been undertaken. A five layer test structure has been designed and used to fabricate wafers with five layers of metallization. Aluminum was found to be unacceptable due to the large hillocks that form even at the lowest annealing temperatures. Sputtered aluminum copper reduces the hillock growth but is undesirable from a plasma etching standpoint the copper halides are not volatile compounds. One possible solution is the use of aluminum with other alloys such as titanium. Films of aluminum/titanium were found to be generally quite smooth after annealing, but our current results show that there are possibly problems with occasional large hillocks, possibly due to residual stress in the aluminum films.

In order to compose a five layer structure, step coverage had to be addressed. It was found that by planarizing the oxide using a plasma process which etches SiO_2 at the same rate as a photoresist overcoat, problems with step coverage were eliminated. An added feature was that the resistance of the interconnects decreased as compared to metallization done over non-planarized surfaces, even when there was only one layer of metal underneath.

The planarization of the oxide was achieved by first depositing two microns of CVD silicon dioxide at 380 degrees centigrade. Then a micron of resist was spun on the oxide and superbaked. Finally the complete structure is plasma etching back for two microns.

The etch rate of the resist is adjusted by varying the oxygen gas flow until it is equal to the rate of etch of the oxide. The result is planarized oxide. Cross sectional photographs reveal that the resist does indeed planarize the surface for short distances, but not for long distances. This is not important for step coverage though because it is the short distances that are important. This is because step coverage is a problem only over steep steps. In addition, the planarity of the surface is as good as what one can achieve using polyimides.

Another experiment which was tried was a comparison of CVD silicon dioxide to oxide deposited using plasma enhancement (which is thought to give more conformal coverage over steps). The step coverage was better on the planarized oxide. Also, electrical measurements revealed that the resistance of aluminum/copper interconnects deposited on both types of surfaces was significantly lower on the planarized CVD oxide surfaces.

Hillocks are another important problem to solve. As can be seen in the surface profilometer plots, evaporated aluminum exposed to 380 degrees (temperature used in depositing oxide) results in hillocks (and voids) each as much as 0.5 μm high. This was reduced by adding copper to the aluminum, but not eliminated. The use of aluminum/titanium was found to eliminate hillocks almost completely. This film is deposited by alternately depositing 50 Angstroms of aluminum with 4 Angstroms of titanium. The alloy formed is Al_3Ti which has a melting point twice that of aluminum. The only problem to be resolved is the appearance of large particles (or possibly hillocks) made of Al/Ti.

There are many reasons for studying five layers of interconnections. One might want to use additional layers of interconnects for such things as power and ground. By dedicating a layer to such things, problems such as electromigration can be reduced and such things as ground planes can be made. In addition, with the advent of silicon compilers (and their routers) and ULSI (ultra large scale integration), a need for more levels of interconnections is developing (vertical integration). Another possible use is to provide interconnections between dies for wafer scale integration.

Staff: K. C. Saraswat, J. P. McVittie, D. G. Gardner, T. Michalka.

Related Efforts: Trotter (Miss. State.).

References: [Saraswat-WCVD 84].

4.4 Cell Library

The Cell Library is finally available as a book from Addison-Wesley and in machine readable form from Addison-Wesley (or via ARPAnet for DARPA VLSI research groups). The book is in full color, and contains considerable additions beyond the July '81 version of the Cell Library. Many thanks to the people who contributed cells, who helped with testing them, who participated in the massive job of documenting them, and who encouraged us along the way.

Staff: R. Mathews, J. Newkirk, C. Burns, and everyone else on the project.

References: [newkirkLibrary83 83]

4.5 Design Classes

The Stanford Mead/Conway design classes have continued to evolve. Starting in the fall of '82, they became a 3-quarter sequence. The first quarter is the introductory class, but with a paper-only design project. This format allows TAs and graders to handle most of the routine work, important since 125 students took the class in Fall Quarter '82, 60 took it in Spring Quarter '83, and 125 are taking it this Fall Quarter. The second and third quarters are the design and testing laboratories, with a more manageable enrollment of about 50 students.

In the '82 sequence, because the students now had an entire quarter to do a design and because they had the prior experience of a first paper design, we allowed the projects to grow as large as their designers desired. That was a mistake, but the results were dramatic \- half of the projects contained over 10,000 transistors. We provided no significant new tools except a simple channel router and ICDEBUG. The end-quarter rush of design-rule checking and simulation overwhelmed our VAX, so checking was no more thorough than in previous years, and the testing results were very similar.

A small number of teams undertook bulk CMOS projects. We provided plotting, design-rule checking, RSIM, and a small library of I/O pads and pieces of a precharged PLA. Since the Stanford CMOS process is n-well and the MOSIS CMOS process is p-well, all designs were described in psuedo-twin-well design rules. (These rules are fully symetric, with explicit well and shorting layers of both types.) The resulting designs were then mapped to each target process and submitted for fabrication. As of this writing, we have received apparently good chips from the Stanford IC Lab run, but we have only tested them partially. A MOSIS CMOS run returned this week.

The last four years of design classes at Stanford are summarized in [mathewsTRtwo83 83]. This technical report begins with a short paper describing the class from the instructor's perspective, but it is mostly a picture book of abstracts and plots displaying almost all of the class designs carried through since the second design class in the spring of 1980. A limited number of copies are available for distribution; contact Rob Mathews (rob%helens@score).

Staff: J. Newkirk, R. Mathews, T. Saxe, S. Taylor

References: [mathewsTRtwo83 83]

References

[BaskettBechtolsheim 82]

Bechtolsheim, A., Baskett, F., and Pratt, V.
The SUN Hardware Architecture.
Technical Report 229, Computer Systems Laboratory, Stanford
University, March, 1982.

[Brown 83]

Brown, H., Tong, C. and Foyster, G.
Palladio: An Exploratory Environment for Circuit Design.
Computer, December, 1983.
To be published.

[ElGamalGreene 83]

El Gamal, A., Greene, J., and Pang, K.
VLSI Complexity of Coding.
1983.
submitted to IEEE Trans. on Infor. Theory.

[Goodwin-Maui 83]

S. H. Goodwin and J. D. Plummer.
Trench Isolation Technology and Device Physics.
In *1983 Symposium on VLSI Technology*, pages 28-29. IEEE Electron
Devices Society and Japan Society of Applied Physics, September,
1983.

[Greene 83]

Greene, J.
Configuration of VLSI Arrays in the Presence of Defects.
PhD thesis, Stanford University, December, 1983.

[GreeneElGamal 83]

Greene, J. and El Gamal A.
Configuration of VLSI Arrays in the Presence of Defects.
1983.
submitted to J. ACM.

[GrossThomas 83]

Gross, T.
Code Optimization Techniques for Pipelined Architectures.
In *Proc. Compcon Spring 83*, pages 278 - 285. IEEE Computer Society,
San Francisco, March, 1983.

[Hennessy 81]

Hennessy, J.L.
A Language for Microcode Description and Simulation in VLSI.
In Seitz, C. (editor), *Proc. of the Second CalTech Conference on VLSI*,
pages 253-268. California Institute of Technology, January, 1981.

- [Hennessy 83] Hennesst, J.L.
Partitioning Programmable Logic Arrays.
In *Proc. Int. Conf. on Computer-Aided Design*. IEEE, Santa Clara,
Ca., September, 1983.
- [HennessyGross 83]
Hennessy, J.L. and Gross, T.R.
Postpass Code Optimization of Pipeline Constraints.
ACM Trans. on Programming Languages and Systems 5(3), July,
1983.
- [HennessyJouppi 81]
Hennessy, J.L., Jouppi, N., Baskett, F., and Gill, J.
MIPS: A VLSI Processor Architecture.
In *Proc. CMU Conference on VLSI Systems and Computations*, pages
337-346. Computer Science Press, October, 1981.
- [HennessyJouppi 83]
Hennessy, J., Jouppi, N., Przybylski, S., Rowen, C. and Gross, T.
Design of a High Performance VLSI Processor.
In *Proc. of the Third Caltech Conf. on VLSI*. Calif Institute of
Technology, Pasadena, Ca., March, 1983.
- [Hochschild 83] Hochschild, P., Mayr, E., and Siegel, A.
Techniques for Solving Graph Problems in Parallel Environments.
1983.
to appear in Twenty-fourth FOCS Proceedings, Nov., 1983.
- [Jouppi 83a] Jouppi, N.
TV: An nMOS Timing Analyzer.
In Randal Bryant (editor), *Proceedings 3rd CalTech Conference on
VLSI*. California Institute of Technology, March, 1983.
- [Jouppi 83b] Jouppi, N.
Timing Analysis for nMOS VLSI.
In *Proceedings 20th Design Automation Conference*, pages 411-418.
1983.
- [Karlin 83] Karlin, A. R., Trickey, H. W., and Ullman, J. D.
Experience With a Regular Expression Compiler.
Technical Report STAN-CS-83-972, Stanford Univ., Dept. of CS, June,
1983.
- [Kuo-ISSCC 84] J. B. Kuo, O-H. Kwon, D. C. Galbraith, F-C. Shone, J. D. Shott, J. T.
Walker, R. W. Dutton, and J. D. Meindl.
A 2 μ m Poly Gate CMOS Analog/Digital Array.
In *1984 ISSCC Digest of Technical Papers*. IEEE, February, 1984.
To be published.

[lavaCOMPCON 82]

Mathews, R, Newkirk, J, Eichenberger, P.

A Target Language for Silicon Compilers.

In , pages 349-353. Proceedings of COMPCON, Spring, 1982.

[Lewyn-ISSCC 84]

L. Lewyn and J. D. Meindl.

Scaling Limits of DRAMs.

In *1984 ISSCC Digest of Technical Papers*. IEEE, February, 1984.

To be published.

[mathewsTRtwo83 83]

R. Mathews, J. Newkirk.

Introduction to VLSI Systems at Stanford.

Technical Report, Information Systems Lab, EE Dept., Stanford

University, September, 1983.

[Moslehi-Maui 83]

M. M. Moslehi and K. C. Saraswat.

Electrical Characteristics of Devices Fabricated with Ultrathin

Thermally Grown Silicon Nitride and Nitroxide Gate Insulators.

In *1983 Symposium on VLSI Technology*, pages 92-93. IEEE Electron Devices Society and Japan Society of Applied Physics, September, 1983.

[newkirkLibrary83 83]

J. Newkirk, R. Mathews.

The VLSI Designer's Cell Library.

Addison-Wesley, Reading MA, 1983.

[Ossher 83]

Harold L. Ossher and Brian K. Reid.

FABLE: A Programming Language Solution to IC Process Automation Problems.

In *Proceedings of the SIGPLAN '83*. ACM, June, 1983.

To be published.

[Pfiester-ISSCC 84]

J. R. Pfiester, J. D. Shott, and J. D. Meindl.

Scaling Limits of NMOS and CMOS Digital Circuits.

In *1984 ISSCC Digest of Technical Papers*. IEEE, February, 1984.

To be published.

[Pfiester-Maui 83]

J. R. Pfiester, J. D. Shott, and J. D. Meindl.

E/D CMOS -- A High Speed VLSI Technology.

In *1983 Symposium on VLSI Technology*, pages 44-45. IEEE Electron Devices Society and Japan Society of Applied Physics, September, 1983.

- [Przybylski 83] Przybylski, S.
The Design Verification and Testing of MIPS.
In *Proceedings of Conference on Advanced Research in VLSI*. MIT,
January, 1983.
to appear.
- [Saraswat-WCVD 84]
K. C. Saraswat, S. Swirhun, and J. P. McVittie.
Selective CVD of Tungsten for VLSI Technology.
In *165th Meeting of the Electrochemical Society*. The Electrochemical
Society, May, 1984.
To be published.
- [Ullman 83] Ullman, J. D.
A Communication-Time Tradeoff.
1983.
unpublished memorandum, Stanford Univ., Dept. of CS.
- [wolfCIT83 83] W. Wolf, J. Newkirk, R. Mathews, R. Dutton.
Dumbo, a Schematic-to-Layout Compiler.
In *Proc. 3rd CalTech Conf. on VLSI*. CalTech, March, 1983.
- [wolfICCAD83 83]
W. Wolf, R. Mathews, J. Newkirk, R. Dutton.
Two-Dimensional Compaction Strategies.
In *Proc. International Conf. on Computer-aided Design*. IEEE,
September, 1983.
- [Yan 83] Yan, J., Foyster, G. and Brown, H.
*An Expert System for Assigning Mask Levels to Interconnect in
Integrated Circuits*.
Technical Report HPP-83-39, Stanford University Heuristic
Programming Project, 1983.

Publications

- [Brown 83] Brown, H., Tong, C. and Foyster, G.
Palladio: An Exploratory Environment for Circuit Design.
Computer, December, 1983.
To be published.
- [ElGamalGreene 83]
El Gamal, A., Greene, J., and Pang, K.
VLSI Complexity of Coding.
1983.
submitted to IEEE Trans. on Infor. Theory.
- [Goodwin-Maui 83]
S. H. Goodwin and J. D. Plummer.
Trench Isolation Technology and Device Physics.
In *1989 Symposium on VLSI Technology*, pages 28-29. IEEE Electron
Devices Society and Japan Society of Applied Physics, September,
1983.
- [GreeneElGamal 83]
Greene, J. and El Gamal A.
Configuration of VLSI Arrays in the Presence of Defects.
1983.
submitted to J. ACM.
- [Hennessy 83] Hennesst, J.L.
Partitioning Programmable Logic Arrays.
In *Proc. Int. Conf. on Computer-Aided Design*. IEEE, Santa Clara,
Ca., September, 1983.
- [HennessyGross 83]
Hennessy, J.L. and Gross, T.R.
Postpass Code Optimization of Pipeline Constraints.
ACM Trans. on Programming Languages and Systems 5(3), July,
1983.
- [HennessyJouppi 83]
Hennessy, J., Jouppi, N., Przybylski, S., Rowen, C. and Gross, T.
Performance Issues in VLSI Processor Design.
In *Proc. Int. Conf. on Computer Design*. IEEE, Rye, N.Y., October,
1983.
- [Hochschild 83] Hochschild, P., Mayr, E., and Siegel, A.
Techniques for Solving Graph Problems in Parallel Environments.
1983.
to appear in Twenty-forth FOCS Proceedings, Nov., 1983.

- [Jouppi 83] Jouppi, N.
Timing Analysis for nMOS VLSI.
In *Proceedings 20th Design Automation Conference*, pages 411-418.
1983.
- [Karlin 83] Karlin, A. R., Trickey, H. W., and Ullman, J. D.
Experience With a Regular Expression Compiler.
Technical Report STAN-CS-83-972, Stanford Univ., Dept. of CS, June,
1983.
- [Kuo-ISSCC 84] J. B. Kuo, O-H. Kwon, D. C. Galbraith, F-C. Shone, J. D. Shott, J. T.
Walker, R. W. Dutton, and J. D. Meindl.
A 2 μ m Poly Gate CMOS Analog/Digital Array.
In *1984 ISSCC Digest of Technical Papers*. IEEE, February, 1984.
To be published.
- [Lewyn-ISSCC 84] L. Lewyn and J. D. Meindl.
Scaling Limits of DRAMs.
In *1984 ISSCC Digest of Technical Papers*. IEEE, February, 1984.
To be published.
- [Moslehi-Maui 83] M. M. Moslehi and K. C. Saraswat.
Electrical Characteristics of Devices Fabricated with Ultrathin
Thermally Grown Silicon Nitride and Nitroxide Gate Insulators.
In *1983 Symposium on VLSI Technology*, pages 92-93. IEEE Electron
Devices Society and Japan Society of Applied Physics, September,
1983.
- [Pfiester-ISSCC 84] J. R. Pfiester, J. D. Shott, and J. D. Meindl.
Scaling Limits of NMOS and CMOS Digital Circuits.
In *1984 ISSCC Digest of Technical Papers*. IEEE, February, 1984.
To be published.
- [Pfiester-Maui 83] J. R. Pfiester, J. D. Shott, and J. D. Meindl.
E/D CMOS -- A High Speed VLSI Technology.
In *1983 Symposium on VLSI Technology*, pages 44-45. IEEE Electron
Devices Society and Japan Society of Applied Physics, September,
1983.

[Saraswat-WCVD 84]

K. C. Saraswat, S. Swirhun, and J. P. McVittie.

Selective CVD of Tungsten for VLSI Technology.

In *165th Meeting of the Electrochemical Society*. The Electrochemical Society, May, 1984.

To be published.

[Singh-IEDM 83] H. J. Singh, K. C. Saraswat, J. D. Shott, J. P. McVittie, and J. D. Meindl.

Scaling of SOI/PMOS Transistors.

In *1989 IEDM Digest of Technical Papers*. IEEE Electron Devices Society, December, 1983.

To be published.

[sucarATG83 83] H. Sucar.

PLA Redundancies Filter and Test Pattern Generator.

September 1983.

VLSI File #092083.

[torresICDEBUG83 83]

J. Torres.

ICDEBUG Manual.

1983.

[Ullman 83]

Ullman, J. D.

A Communication-Time Tradeoff.

1983.

unpublished memorandum, Stanford Univ., Dept. of CS.

[watsonCHERRY82 82]

Watson, I., Newkirk, J., Mathews, R. and Boyle, D.

ICTEST: A Unified System for Functional Testing and Simulation of Digital ICs.

In *Proc. Cherry Hill International Test Conference*. Philadelphia, 1982.

[wolfCIT83 83] W. Wolf, J. Newkirk, R. Mathews, R. Dutton.

Dumbo, a Schematic-to-Layout Compiler.

In *Proc. 3rd CalTech Conf. on VLSI*. CalTech, March, 1983.

[wolfICCAD83 83]

W. Wolf, R. Mathews, J. Newkirk, R. Dutton.

Two-Dimensional Compaction Strategies.

In *Proc. International Conf. on Computer-aided Design*. IEEE, September, 1983.

[Yan 83]

Yan, J., Foyster, G. and Brown, H.

An Expert System for Assigning Mask Levels to Interconnect in Integrated Circuits.

Technical Report HPP-83-39, Stanford University Heuristic Programming Project, 1983.

END

FILMED

12-83

DTIC